*RveLinkedList.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks

<global>

CLList.c × *RveLinkedList.c × Merge2List.c ×

```c
#include<stdio.h>
#include<stdlib.h>

typedef struct node
{
    int data;
    struct node *next;
}nt;
nt* insert(nt *r)
{
    nt *p=NULL;
    p=(nt*)malloc(sizeof(nt));
    if(p!=NULL)
    {
        printf("enter data ");
        scanf("%d",&p->data);
        if(r==NULL)
            r=p;
        else
        {
            r->next=p;
            r=p;
        }
        r->next=NULL;
    }
return r;
```

Search

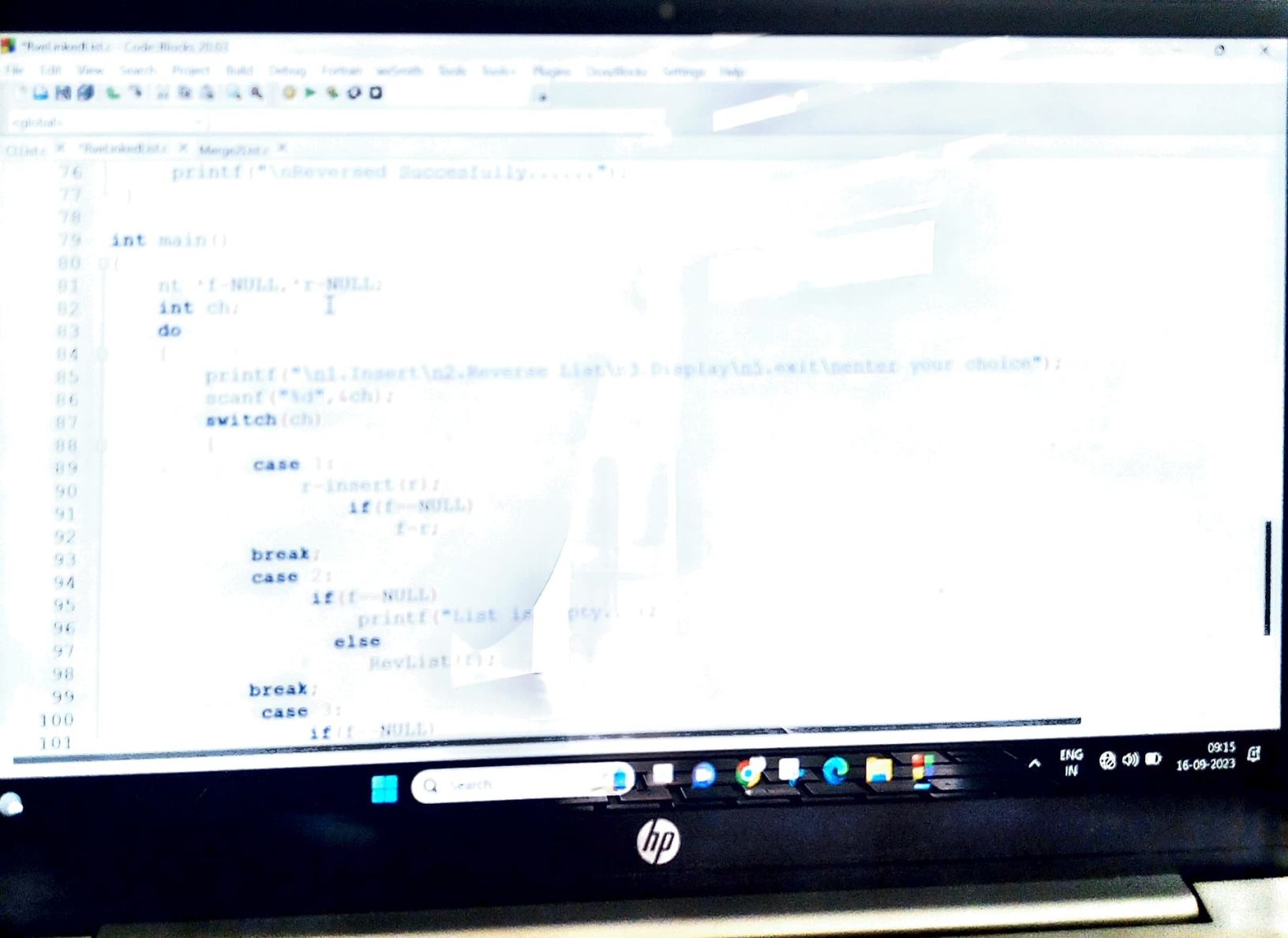
RveLinkedList.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

CLList.c × *RveLinkedList.c × Merge2List.c ×

```
52
53  void RevList(nt *f)
54  {
55    nt *s=NULL,*last=NULL;
56      int t=0,i=1,j=0,k=1;
57         j=len(f);
58          s= last = f;
59          while(i<j)
60           {
61                  k=1;
62                  while(k<j)
63                  {
64                       last=last->next;
65                        k++;
66                  }
67                  //swaping first and last 2nd and 2nd last and so on.....
68                  t = s->data;
69                  s->data = last->data;
70                  last->data = t;
71
72              i++,j--;
73              s=s->next;
74              last = f;  //again assigning address of first node
75           }
76       printf("\nReversed Succesfully.....");
77  }
```


printf("\nReversed Successfully.....");
int main()
nt *f=NULL,*r=NULL;
int ch;
do
printf("\n1.Insert\n2.Reverse List\r) D:splay\n3.exit\nenter your choice");
scanf("%d",&ch);
switch(ch)
case 1:
r=insert(r);
if(f==NULL)
f=r;
break;
case 2:
if(f==NULL)
printf("List is pty.");
else
RevList(f);
break;
case 3:
if(f==NULL)
ENG
IN
09:15
16-09-2023

<global>

CLList.c × *RveLinkedList.c × Merge2List.c ×

```
87          switch(ch)
88          {
89              case 1:
90                  r=insert(r);
91                      if(f==NULL)
92                          f=r;
93              break;
94              case 2:
95                  if(f==NULL)
96                      printf("List is empty..");
97                    else
98                       RevList(f);
99              break;
100              case 3:
101                  if(f==NULL)
102                      printf("List is empty..");
103                    else
104                       display(f);
105             }
106     }while(ch<=3);
107
108
109 return 0;
110 }
111
```

```
37          }
38      }
39
40  node* Merge(node *first, node *second)
41  {
42          node *third=NULL,*last;
43
44          if (first->data < second->data)
45          {
46              third = last = first;
47              first = first->next;
48              //last->next = NULL;
49          }
50          else
51          {
52              third = last = second;
53              second = second->next;
54             // last->next = NULL;
55          }
56
57          while (first != NULL && second != NULL)
58          {
59              if (first->data < second->data)
60              {
61                  last->next = first;
62                  last = first;
```

Merge2List.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Set

<global> CreateList(node* last, int n) : node

CLList.c *RveLinkedList.c Merge2List.c

```
58          {
59              if (first->data < second->data)
60              {
61                  last->next = first;
62                  last = first;
63                  first = first->next;
64                 // last->next = NULL;
65              }
66              else
67              {
68                  last->next = second;
69                  last = second;
70                  second = second->next;
71                 // last->next = NULL;
72              }
73      }
74
75      if (first != NULL)
76          last->next = first;
77      else
78          last->next = second;
79
80  return third;
81  }
82
83  int main()
```